An Impact Assessment Model for Distributed Adaptive Security Situation Assessment[*]

Mark Heckman, Nikhil Joshi, Marcus Tylutki, Karl Levitt
University of California, Davis

James Just, Lawrence Clough
Teknowledge Corporation

Abstract: The goal of any intrusion detection, anti-virus, firewall or other security mechanism is not simply to stop attacks, but to protect a computing resource so that the resource can continue to perform its function. A computing resource, however, is only a component of a larger system and mission. Sometimes, the efforts made to stop an attack on a resource may be as bad as the attack itself in terms of affecting the overall ability of the system to complete its mission. What is needed is a method of choosing responses to attacks on components that still allows the system to achieve its goals. We present a model of computing resources and of how the loss or degradation of resources impacts the ability of a system to complete its mission. A human or robot analyst can use the model to assess the security status of a monitored system and to allocate resources in an optimal way.

1. Introduction

Intrusion detection and other computer system security mechanisms are primarily focused on detecting attacks on computing resources. Some mechanisms, such as anti-virus software and firewalls, are also good at blocking certain types of attacks. A computing resource, however, is usually a component in a larger system. A proper response to an attack must take into account not only the effects of the attack on a particular resource, but also the affect of the attack and the response on the overall system. The primary goal of any security response must be to preserve the ability of the overall system to perform its function, not simply to protect particular resources.

Currently, a disproportionate amount of human effort is required to identify a widespread, distributed cyber attack and to characterize it sufficiently to formulate an effective response. It is difficult to distinguish components of such an attack (intrusions, attacks, or precursors to the main attack) from events that have only local significance. This is because widespread coordinated activities cannot be identified from local data alone: correlating and identifying such activities requires human reasoning and expertise, and configuring sensors to confirm hypothesized attacks requires manual intervention.

The objective of the work presented here is to provide a situation assessment capability to assist analysts in understanding the overall functional and security status of a system and the implications with respect to the ability of the system to accomplish its goals. We present a model and system design for automatically collecting data, analyzing the impact of events on a system, and presenting the resulting situation assessment in a succinct and easily understandable format. A human or automated analyst can use the system to test hypotheses and to develop an optimal response strategy.

An example of an intrusion response system is IDIP (Intrusion Detection and Isolation Protocol) [6]. IDIP correlates sensor data to detect attacks and takes steps to isolate the attacker from the network (e.g., using firewall rules, breaking links, etc.). In some cases, however, the retaliation may not be the best response (e.g., if a critical system is on the cut-off section). Whereas IDIP is focused on the security state of the resources, our system is focused on optimizing the dynamic security and functional state of the system as a whole.

Another project that has similarities to our own work is the cost sensitive modeling of Wenke

[*] This research was sponsored by DARPA, research contract number F30602-99-C-0138, through the Air Force Research Laboratory. The results contained in this paper represent the views of the authors and not those of DARPA or the Air Force Research Laboratory.[1]

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2005**	2. REPORT TYPE	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **An Impact Assessment Model for Distributed Adaptive Security Situation Assessment**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defense Advanced Research Projects Agency,3701 North Fairfax Drive,Arlington,VA,22203-1714**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		**10**	
unclassified	**unclassified**	**unclassified**			

Lee [4]. That work describes a model that evaluates the major cost factors associated with security administration, including development cost, operational cost, damage cost due to attacks, and the cost of manual or automated response. Their model creates an attack taxonomy to evaluate the impact of attacks on the damage and response costs. For example, events with a higher response cost than damage cost maybe ignored or events that have a higher operational cost to detect than damage cost may not even be scanned for. Unlike the cost sensitive model, our model supports swapping resources and reconfiguring a system, as well as relating resources to critical mission time and quality constraints.

2. Scenario

In this section we describe an example scenario that provides the motivation for our model.

In the U.S. military, a *Task Force* is an ad hoc organization of specialized units that is established by an authority to accomplish a specific mission. A *Joint Task Force* (JTF) involves military units that "belong" to two or more service branches. The organization of a JTF is usually hierarchical to allow for various levels of leadership and abstraction. Each mission is structured according to a plan into a series of time phased, often hierarchical tasks. The plan represents the assigning authority's best estimate of how to accomplish their objectives given what they know of the situation and their constraints. These tasks are assigned to individual units, with the lowest level tasks representing actual execution tasks. The tasks are usually phrased in terms of the objectives to be achieved so as to give the executing units the maximum flexibility in how they accomplish the tasks. We refer to the lowest level of physically and electronically protected networked environment (e.g., a LAN and its computers) as an *enclave*.

As an example, consider a JTF whose overall mission it is to collect and deliver food and other supplies to the International Relief Organization (IRO) in the nation of Tamboria. The JTF collects the supplies in the United States, flies them to Tamborian International Airport in Tamboria, and then trucks the supplies to distribution points run by the IRO. One of the enclaves in the JTF is called "DLA Forward". DLA Forward has the mission of developing detailed instructions for trucking the supplies from the Tamborian International Airport to the IRO distribution points. The information product from DLA Forward is the set of trucking instructions – the number, type, and size of the trucks that are needed, their schedules, driving directions, etc. – that is sent in this example to JTF headquarters in the capital of Tamboria.

The overall mission of the JTF is to collect, ship by air, and deliver the cargo to the IRO in Tamboria, so the information product of DLA forward is a component of the overall organizational mission. The cargo includes food and other perishables, so DLA Forward is required to deliver its trucking instructions within a certain time frame. Also important to the success of the mission is the *quality* of the trucking instructions calculated by DLA Forward. If the instructions are bad, the number or type of trucks sent to transport the supplies will be wrong, so the supplies may not be delivered on time. The time limits and minimum quality standards constitute the *constraints* on DLA Forward's mission.

The mission of DLA Forward is to calculate the trucking instructions. The mission tasks are 1) to receive the airplane cargo manifests, 2) to convert the manifests into trucking instructions, and 3) to transmit the trucking instructions to the JTF headquarters. Each of the mission tasks uses computing resources: Receiving the cargo manifests uses a communication link, converting the manifests into trucking instructions uses a computer or other calculation method, and transmitting the instructions uses another communication link. These tasks are sequential: DLA Forward must first receive the manifests before it can calculate a plan, and the plan must be calculated before it can be transmitted.

DLA Forward's mission is depicted in figure 1. In the figure, mission tasks are represented as rectangles and computing resources are represented by ovals. Arrows depict information flow. Lines between resources and tasks represent what resources are available to complete a task. The receive task is labeled *comm-in*, the calculation task is labeled *comp-power*, and the transmission task is labeled *comm-out*. The manifests are themselves the outputs of another mission, assigned to a different enclave. The dashed box labeled manifests represents that mission. Similarly, the

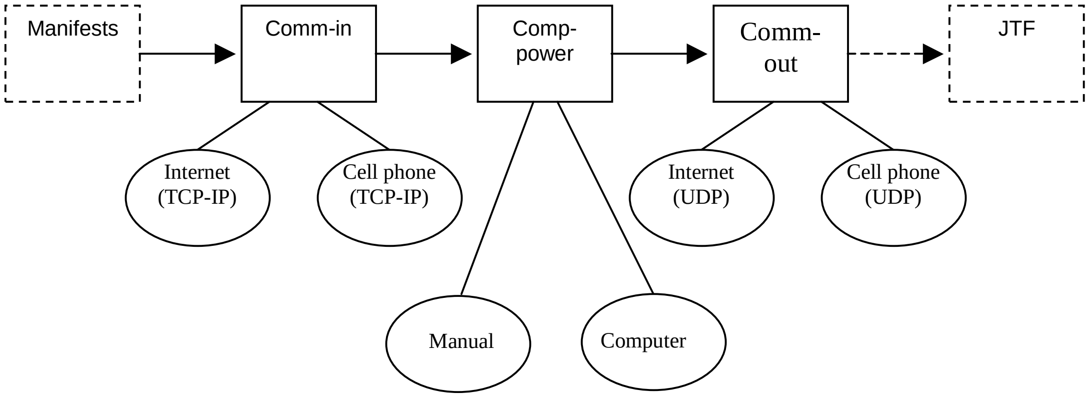


Figure 1 – Model of DLA Forward Mission Tasks and Resources

dashed box labeled *JTF* represents the JTF's task to receive the trucking instructions.

As shown in the figure, each of the tasks has two alternative resources that can be used to accomplish the task. The input communications link used to receive the cargo manifest can be either a 56K modem-based TCP/IP connection (the oval labeled *internet*) or a 40K wireless modem connection (*cell phone*) that also uses TCP/IP. The computing resource used to calculate the trucking plan can be either a special program running on a PC (*computer*), or the same calculation can be done by hand (*manual*). The output communications link can be either via a UDP connection on the 56K modem (*internet*) or via a UDP connection on the 40K wireless modem (*cell phone*). For each resource, either alternative may be used to achieve the same goal, but some alternatives may be better than others in terms of time and quality. For example, the trucking plan calculation using the special program running on a PC is likely to be completed much more quickly and more accurately than the same calculation performed by hand.

The infrastructure in Tamboria is antiquated, at best, and often subject to disruptions and outages. At times, for example, the heat and humidity may bring down the computer resource used to calculate the trucking instructions, forcing the enclave staff to use pencil and paper methods to perform the calculations. Or the poor quality phone lines may slow the Internet connections to a crawl or cut them off entirely. Or it is conceivable that someone hostile to the effort to aid Tamboria may attempt to disrupt communications and the computer through denial of service or other attacks.

We are interested in assessing the impact of attacks, disruptions, and outages on the ability of the DLA Forward enclave to complete its mission. When one resource is degraded then its alternative can be pressed into service, but at what point does an alternative resource become preferable? And if key resources become unavailable, will the enclave still be able to complete its mission or must JTF headquarters find an alternative enclave to carry out the mission? We must be able to model the different resources and the effects of disruptions in order to determine an optimal recovery strategy.

3. Model

The model must represent the following components:

- Missions and tasks (i.e., outputs) and their assignments to units and enclaves
- Inputs required for the mission tasks
- Time and quality mission constraints
- Resources that may be used by a unit or enclave to accomplish its mission
- Resource attributes (e.g., time and quality)
- Attacks or other resource disruptions

Our entire system model consists of a hierarchical resource and task graph model, a calculation model, and a human or expert system analyst.

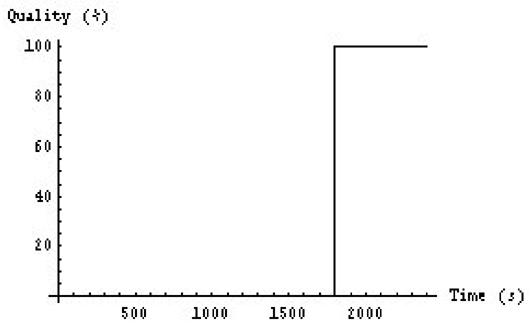


Figure 2 – Computer TQ Graph

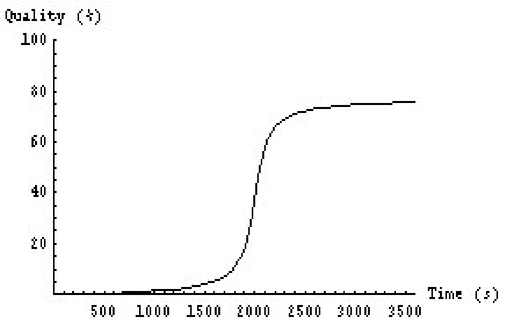


Figure 3 – Manual TQ Graph

3.1. Hierarchical Resource and Task Graph Model

An example of our resource and task graph model is shown in figure 1, above. Rectangles represent tasks, ovals represent resources, and arrows depict information flow. The output of one task is the input to another. Our model is hierarchical: the mission of DLA Forward, for example, can be represented as a single task in a more-abstract graph that represents the mission of the JTF.

Where the outputs of one enclave feed as inputs into another, the boundary between enclaves may seem blurry. For example, if sending the plans is the task of DLA Forward and JTF has the task of receiving the plans, but the two tasks share the same communication channel, how does one determine which enclave is responsible for the transmission? In our scenario, it may be that a client process at JTF contacts a server at DLA Forward to download the manifests. The task of DLA Forward, therefore, would be to run the server, not to perform that actual transmission. The details of assigning tasks to units and enclaves depend on the specific tasks and resources used.

3.2 Calculation Model

We represent resources, tasks, and missions as *anytime algorithms* [3, 5], functions that relate two attributes: Time (*T*) and quality (*Q*).[1] *T* represents the time it takes to accomplish some task – the computation time required to translate a manifest to a trucking plan, for example. *Q* represents the accuracy of the output, usually in the form of a percentage relative to some theoretical maximum of 100%. In our trucking plan example, 100% quality might mean that the instructions result in all trucks that are sent being fully loaded with no materials left at the airport, while 90% quality might mean that, on average, the trucks end up only 90% loaded. The meaning of “quality” depends on the details of the task. Most work on anytime algorithms has assumed monotonically increasing functions of output quality vs. time. The motivation behind this justification is that if a later result is of lower quality than any previous result, the previous higher quality result could be used. [3]

The composition of the functions for resources used to complete a mission gives the function for the mission.

3.2.1 Resource Graphs

The two attributes *T* and *Q* are related: to achieve a higher quality, for example, may require a longer time. The combination of the two attributes and their relationship can be represented as a two-dimensional graph, called a TQ graph. The surface of the graph represents the maximum time it would take to achieve that minimum quality level. Anything inside the shape defined by the graph surface down to the time axis (i.e., points that represent a longer time, lower quality, or a combination of the two) is considered to be achievable using that resource. For example, if a resource can deliver

[1] We also have been considering third and fourth dimensions: reliability and cost. Reliability refers to the likelihood of a resource meeting some time/quality constraints. For example, our comm-in TCP/IP connection, given enough time, can deliver 100% quality (i.e., the entire, uncorrupted manifest) with 100% reliability, but in a very short time, possibly due to a noisy line, we cannot guarantee that level of service. Cost refers to the actual material and monetary costs that the use of a particular resource may entail.

80% quality in 10 minutes, it can also deliver 70% quality in that same time. Similarly, anything outside the shape defined by the graph surface up to 100% quality may be possible, but is not guaranteed.

Resources are represented by TQ graphs, and the TQ graphs for the different resources used to complete a task are composed into a TQ graph for the task. The TQ graph for a task can then be composed with other task graphs to generate a mission TQ graph.

As an example, consider the representation of the computing resources used by DLA Forward. Let us say that the software/hardware that is used to calculate the trucking instructions will complete its calculations in no more than 1800 seconds with 100% quality, but that there are no partial results; i.e., in less than 1800 seconds there will be 0% quality. The TQ graph is therefore a step function, as shown in figure 2.

Making the same calculation manually will probably take longer than using a computer, but there may also be some partial or approximate results (i.e., results of a lower quality) possible in less time than it takes to completely calculate the trucking plans. Manual calculations are also more likely to contain errors than a computer calculation, but given sufficient time to recheck figures, an analyst can probably improve the quality of a manual calculation. These factors are represented in the TQ graph for manual computation shown in figure 3. The figure shows a low quality level up until about 1900 seconds, after which the quality increases steeply until it levels off at 80% quality at about 2500 seconds. As time increases after that point, the quality continues to improve at a slow rate.

There is a separate TQ graph for each resource. These graphs are shown in figures 4 through 7. In our example, the cell phone modems are slower than the regular modems, so they take a longer time to reach an equivalent quality level. Because UDP is generally faster but less reliable than TCP/IP, the comm-out resources rise in quality more steeply than their comm-in counterparts, but flatten before they reach the quality levels achieved by the comm-in line for the same given time.

These graphs are pure inventions intended as examples only. In practice, we expect that graphs for particular resources will need to be empirically generated.

Plan No.	Comm-in	Comp.	Comm-out
1	internet	computer	internet
2	internet	computer	cell phone
3	internet	manual	internet
4	internet	manual	cell phone
5	cell phone	computer	internet
6	cell phone	computer	cell phone
7	cell phone	manual	internet
8	cell phone	manual	cell phone

Table 1 – Possible plans

3.2.2 Plans

A *plan* is a particular allocation of resources to accomplish a mission task. In our DLA Forward example, we have two resource options for each of the three resource types, giving eight possible plans, as shown in table 1.

The resources in our DLA Forward example are used serially, with no overlap: DLA Forward first receives the manifests via the comm-in link, then it calculates the trucking instructions, then it transmits the results via the comm-out link. The total time it takes DLA Forward to complete its mission is the sum of the times it takes to complete each of the mission tasks. For example, if DLA Forward must report the trucking instructions for a particular manifest within 3000 seconds after the manifests become available, we could allocate 500 seconds to receive the message, 2000 seconds to compute it, and 500 seconds to transmit the instructions. We could also allocate just 100 seconds each to receive and transmit, but 2800 to compute the instructions. Our decision as to how to allocate the time among resources is based on which allocation maximizes the overall quality of our results within that time.

3.2.3 Composing TQ Graphs

Each plan can be represented in the form of a TQ graph that is a maximal composition of the TQ graphs of the plan's resources. TQ graphs for serial plans are calculated in the following way:

1. Choose a time step size to use in the calculation. The smaller the time step size, the more accurate the calculated graph, but the calculation will be more computationally intensive.
2. Beginning with a total time duration of zero and repeating for durations of multiples of

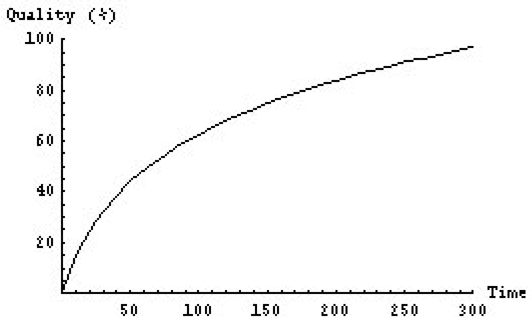


Figure 4 – Internet (TCP/IP)

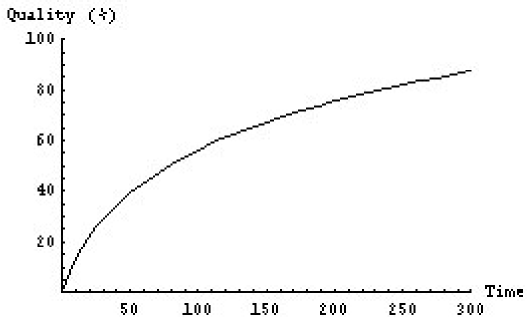


Figure 5 – Cell phone (TCP/IP)

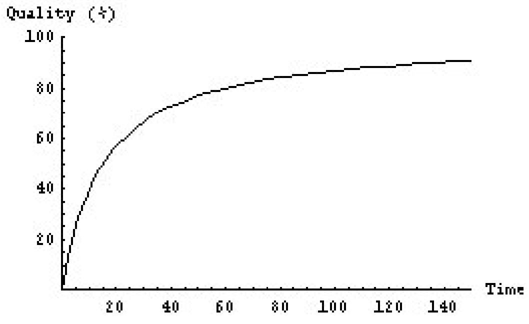


Figure 6 – Internet (UDP)

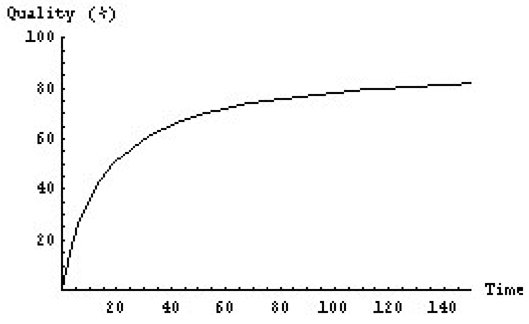


Figure 7 – Cell phone (UDP)

the time step size (up to a maximum allowed time), consider all the possible ways that the time duration can be allocated to each of the resources. For example, there is only one way to allocate zero time duration to all the tasks, there are three ways to allocate one time step among the three resources, there are six ways to allocate two time step units among the three resources, and so on. For *n* resources and *m* time step units, there are C(n+m-1, m) ways to allocate the time units to the resources.

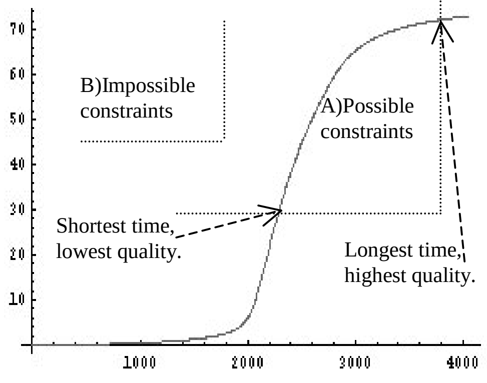


Figure 8 - TQ graph for resource combination 7, showing mission constraints

3. At each time duration, for each possible time allocation find the quality that can be achieved by each resource. To get the composed quality for that particular time allocation, multiply the resource quality percentages (because the tasks are serial and non-overlapping, simple multiplication can be used). For example, if the first task can achieve at best 80% quality, even if the other two tasks can achieve 100% quality, the quality of the results cannot exceed 80%. Out of all of the possible time allocations for that duration, choose the allocation that gives the maximum composed quality. The maximum composed quality for that duration becomes a point on the TQ graph for the plan. For example, at a time duration of three time step units, the best quality can be achieved by allocating one time step unit to each resource (if any resource is allocated no time step units, then it will have a quality of 0% which would make the maximum quality 0% for the plan for that particular allocation).
4. In this way, we construct a TQ graph whose points are the maximum quality possible for

a particular plan at a given time duration. Every point on the graph also represents a particular allocation of time to the set of resources in the plan.

A TQ graph for plan number 7 (cell phone, manual, internet) is shown in figure 8.

It is clear that the computational complexity of the composition algorithm precludes scaling it to larger numbers of combinations. We are currently investigating more efficient algorithms.

3.2.4 Constraints

Each point in the TQ graph for a plan represents the resources that are to be used as well as the time allocated to each resource. For example, if headquarters was willing to accept 30% quality in no more than 3800 seconds (call this "constraint set A"), resource combination number 7 could be used because there are time allocations for the constituent resources that gave acceptable quality within the required time. If, however, headquarters would accept no less than 45% quality but required results within 1800 seconds (call this "constraint set B"), resource combination 7 could not be used because there is no time allocation among the constituent resources that would yield an acceptable level of quality. The constraints on time and quality form a backward "L" shape, with the maximum time constraint on the vertical axis and the minimum quality constraint on the horizontal, as shown in figure 8.

Because there are abstractly an infinite number of points along the part of the graph that satisfies constraint set A, there are abstractly an infinite number of time allocations that we could choose. Under these circumstances, how do we choose a time allocation? The answer depends on which constraint we weigh most heavily. If time is of the essence, we might choose the allocation that gives us the least time. If quality were the most important criteria, we would choose the allocation that gave us the highest quality within the time constraints.

TQ graphs for all plans are shown in figure 9. From top to bottom (highest quality to lowest quality), they are combination numbers 1, 5, 2, 6, 3, 7, 4, and 8. Combination numbers 1,5,2, and 6 all use the computer to create the trucking instructions, which accounts for why they all have higher quality at lower times than 3, 7, 4, and 8, which all use manual calculation. The graphs within each of the pairs (1,5), (2,6), (3,7) and (4,8) converge over time. This occurs because the quality differences between the

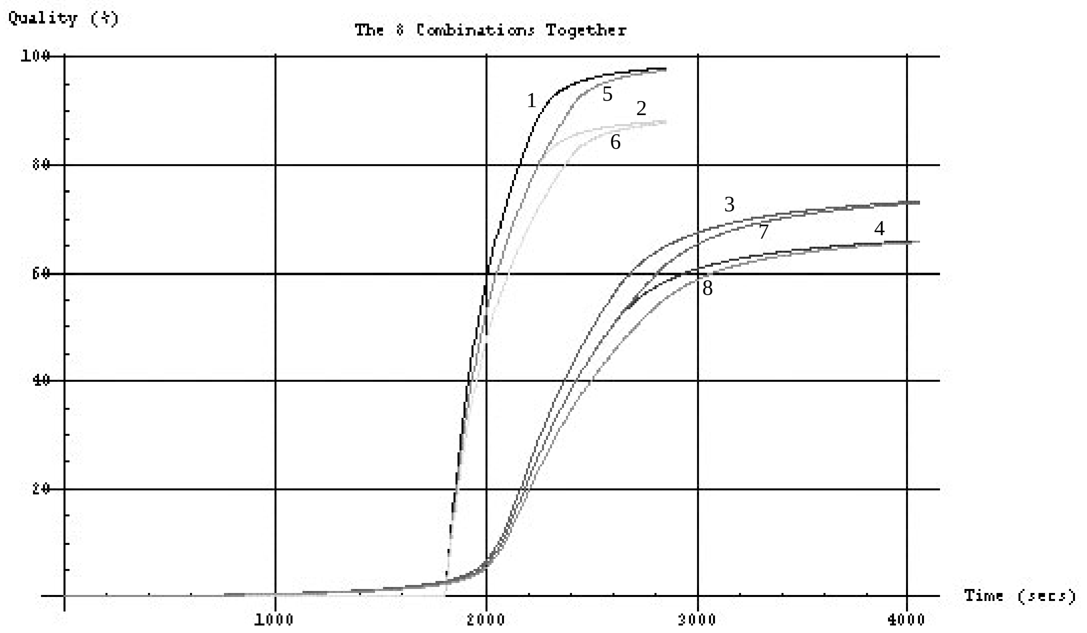


Figure 9 – TQ graphs for all eight resource combinations

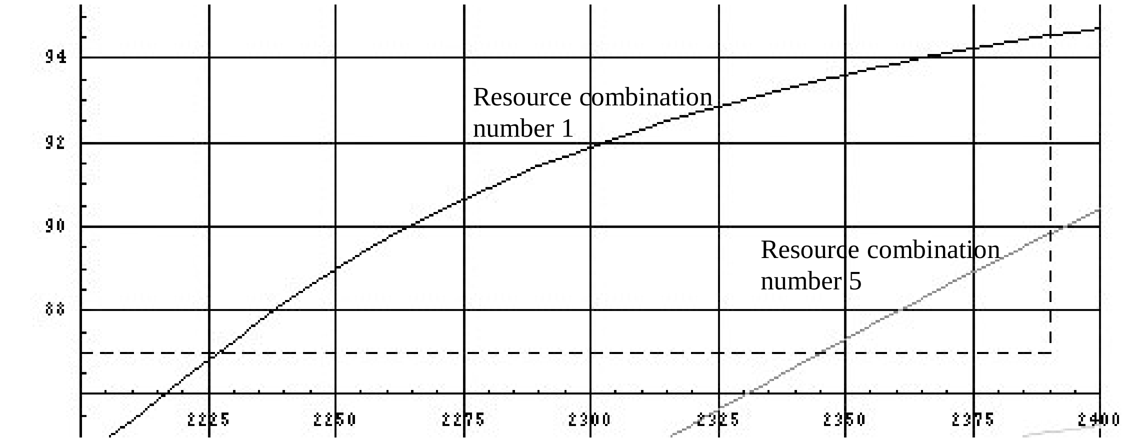


Figure 10 – Multiple resource combinations satisfying constraints

modems and the cell phones disappear as more time is allowed to transmit the plans.

Let us say that the time and quality constraints on DLA Forward's mission are 2390 seconds and 87%, respectively. In this case, two different resource combinations will satisfy the constraints (this is shown in figure 10). How do we choose which resource combination to use? Which plan is chosen depends on the time and quality mission constraints (and probably on other considerations that are not modeled). In this case, resource combination 1 yields us the best quality for all possible times, so it seems like an obvious choice. If, however, we were using other attributes and constraints, such as cost, and resource combination number 5 offered a lower cost, we might choose resource combination number 5 over resource combination number 1.

3.2.5 Attacks

We represent attacks (as well as malfunctions or other outages) as a reduction in the capability of a resource. This is represented in our model by a shifting of the TQ graph down in one or more dimensions. For example, let us say that a computer virus takes down the DLA Forward computer so that only manual computation is possible. The new graph for the computer would show 0% quality for all possible times. The effect on the TQ graph for DLA Forward would be to drag curves 1, 5, 2, and 6 down to 0% quality, leaving the other curves unchanged.

3.3 System Analyst

By monitoring the effects of attacks or other outages on resources, a human or robot analyst can allocate resources to allow a unit to accomplish its mission. An analyst can also use TQ graphs to estimate the effect of a particular response before committing to it. Various scenarios can be compared until an optimal solution is found.

Analysis of the detectable behavior of a resource may allow the analyst to anticipate attacks on other resources and to take appropriate action before the unit's capabilities are affected. Let us say, for example, that there is always a correlation between the speed of the internet comm-in line and the speed of the internet comm-out line. An analyst, noting the slowdown in the comm-in line, might order tests of the comm-out line or else automatically adjust the TQ graph of the comm-out line by an amount that corresponds to the degradation on the comm-in line.

4. Demonstration

We have applied our model in a prototype situation assessment system. The prototype consists of five components – sensors, expert system, calculation engine, a graphical user interface (GUI), and a TQ graph creation tool (figure 11).

The TQ graph creation tool allows users to define missions, tasks, and resources, and to create an initial TQ graph for each resource. The initial resource TQ graphs are inputs to the calculation engine, which uses them to calculate initial mission TQ graphs for each resource combination.

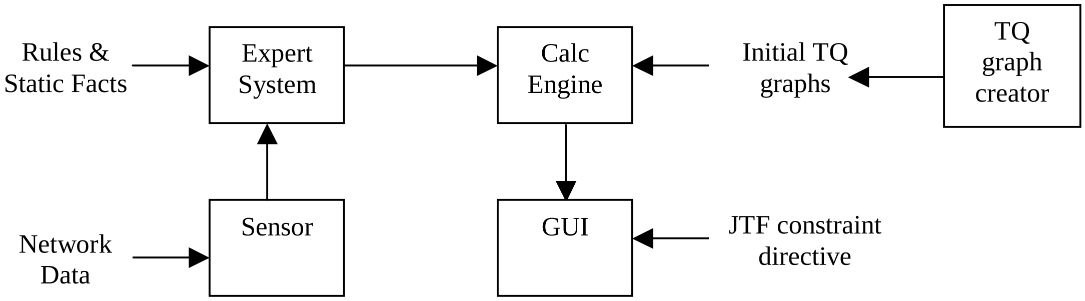


Figure 11 – Prototype design

Sensors track network data and detect anomalous behavior, which is reported to the expert system. The expert system analyzes the sensor data, determines what effect, if any, the reported behavior has on the TQ graphs for the resources, and passes its directives to the calculation engine.

The calculation engine applies the expert system directives to the resource TQ graphs and recalculates the mission TQ graphs for each resource combination. The GUI displays the TQ graphs in real-time as changing conditions affect the resources. An example GUI window is shown in figure 12.

4.1 Expert System

The expert system that we use in our prototype is called JESS (Java Expert System Shell)[2]. The expert system is a Java object that is continually resident inside a wrapper. The wrapper waits for new assertions (i.e., sensor data) and then injects them into the JESS object.

Before beginning normal execution, JESS has an initial fact base and an initial rule base that it reads to initiate the system. The fact base is a list of static facts about each of the computing resources. The rule base is used to translate facts into decisions about whether data reported by the sensors will affect one or more resources, and the degree to which the resources are affected.

When JESS detects that a resource has been affected, it determines the severity of the effect. Our prototype uses integer values to represent the state of each resource. The expert system sends the integer values to the calculation engine.

4.2 Calculation Engine

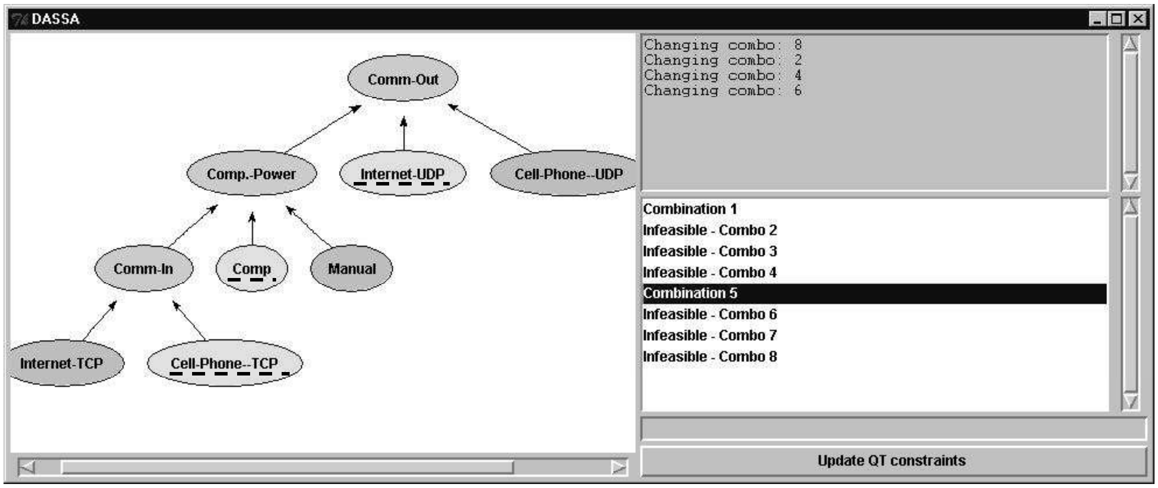
On initialization, the calculation engine reads the initial TQ data file created by the TQ graph creation tool and calculates the original TQ graphs of resource combinations. When the engine receives a directive from the expert system, it is in the form of a resource identifier and a “reliability reduction value” (integer value).

Each TQ graph for a resource is stored as a series of pre-calculated graphs. The integer value part of the expert system directive identifies which of the pre-calculated graphs is to be used to represent the current state of the resource. The pre-calculated graphs are used to simulate the distortion of the resource TQ graphs due to changing conditions. Once new TQ graphs for resources have been identified, the calculation engine calculates the new TQ graphs for each resource combination and sends them to the GUI.

5. Conclusion

Deciding how to respond to attacks on computer resources is a tricky problem. The chief goal is not simply to protect the individual resources, but to preserve the system’s ability to complete its mission. Whatever response is chosen, therefore, must take into account the overall mission of a system and the available resources, as well as data about the attacks.

In this paper we have presented a model for evaluating how attacks on computing resources affect the overall ability of a system to accomplish its mission. The model is based on “anytime algorithms”, which relate time to some measure of output quality. Time/Quality (TQ) graphs represent system resources, and can be



```
Changing combo: 8
Changing combo: 2
Changing combo: 4
Changing combo: 6
```

Combination 1
Infeasible - Combo 2
Infeasible - Combo 3
Infeasible - Combo 4
Combination 5
Infeasible - Combo 6
Infeasible - Combo 7
Infeasible - Combo 8

Figure 12 – Example GUI Window

composed into hierarchical task and mission TQ graphs. The graphs can be used to determine if a system can accomplish its mission (in terms of time and quality constraints).

Our model represents attacks as distortions of resource TQ graphs, which cause the resource to produce lower quality output in a longer time. An analyst who observes that a disruption of resources affects the ability of the system to accomplish its mission can order appropriate responses, which may include stopping the attacks or switching to alternative resources. A prototype system based on our model demonstrates the utility of our approach.

In the future, we intend to experimentally develop TQ graphs for real components and TQ graph distortion parameters for real attacks. Real sensors will be incorporated into our prototype, which will require a better method for translating sensor data into distortion parameters. We also plan to expand the model to include more complex graph topologies that include parallel tasks as well as serial tasks and to develop more efficient algorithms for composing TQ graphs. Similarly, we plan to examine inter-component dependencies and incidents that can be correlated among components. Longer term, we hope to explore additional dimensions, such as cost and reliability. Our model currently deals with issues of integrity and functionality; an interesting and useful extension would be to incorporate confidentiality into the model.

6. Bibliography

[1]http://www.darpa.mil/iso2/cc2/dassasummary.html

[2] JESS – The Java Expert System Shell. Sandia National Laboratories. http://herzberg1.ca.sandia.gov/jess/

[3] S. Zilberstein and S. J. Russell. Optimal Composition of Real-Time Systems. Artificial Intelligence, 82 (1-2):181-213, 1996.

[4] Wenke Lee, Wei Fan, Matt Miller, Sal Stolfo, and Erez Zadok. Toward Cost-Sensitive Modeling for Intrusion Detection and Response. The First ACM Workshop on Intrusion Detection Systems, Athens, Greece, November 2000.

[5] Joshua Grass. Reasoning about Computational Resource Allocation. Crossroads, the ACM Student Magazine, September 1996.

[6] Dan Schnackenberg, Kelly Djahandari, and Dan Sterne. Infrastructure for Intrusion Detection and Response. DISCEX '00 Proceedings, Hilton Head, S.C., January 2000.